



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/715,136

11/18/2003

Sundaram Ramani

MFCP.110234

1912

45809

7590

02/24/2009

SHOOK, HARDY & BACON L.L.P.
(c/o MICROSOFT CORPORATION)
INTELLECTUAL PROPERTY DEPARTMENT
2555 GRAND BOULEVARD
KANSAS CITY, MO 64108-2613

EXAMINER

DAO, THUY CHAN

ART UNIT

PAPER NUMBER

2192

MAIL DATE

DELIVERY MODE

02/24/2009

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/715,136	Applicant(s) RAMANI ET AL.	
	Examiner Thuy Dao	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 02 December 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-9, 11-14, 16-26 and 28-33 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-9, 11-14, 16-26 and 28-33 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 11/18/03 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the amendment filed on December 3, 2008.
2. Claims 1-9, 11-14, 16-26, and 28-33 have been examined.

Response to Amendments

3. In the instant amendment, claims 1, 9, 11, 16, 25, 26, and 28-33 have been amended; claims 10, 15, and 27 have been canceled.
4. The objection to the specification and claim 1, 9, and 33 is withdrawn in view of Applicant's amendments.

Claim Objection

5. Claim 9, lines 1-2, the phrase is considered to read as - "...a computer readable storage [[readable]] medium...- -.

Appropriate correction is requested.

Priority Date

6. In view of persuasive Applicants' arguments, the examiner acknowledges the continuation-in-part priority date (February 28, 2003) of the instant application.

Response to Arguments

7. Applicants' arguments have been considered but are not persuasive.

a) "The rejection of claims 1-33 under 35 U.S.C. § 103(a) as being obvious over Walker, U.S. Publication No. 2004/0015840 in view of Bhansali et al., U.S. Publication No. 2002/0169999 should be withdrawn because the references cannot be combined" (Remarks, pp. 10-16):

The examiner respectfully disagrees with Applicants' assertions. Walker discussed:

"[0003] A technical proposal put forth by Sun Microsystems is described in a document entitled "An XML Data-Binding Facility for the JAVA Platform," by Mark Reinhold of the Core JAVA Platform

Group, published Jul. 30, 1999. In this paper, a schema compiler is proposed that generates JAVA classes from an XML schema. Unfortunately, the use of a schema compiler is problematic and limiting to a developer in terms of control and flexibility of the mapping of JAVA classes to XML.” (emphasis added); and

“[0019] An API in accordance with the invention advantageously avoids the use of a schema compiler to generate new JAVA classes ...” (emphasis added).

As an initial matter, the examiner notes that the Applicants mischaracterized Walker’s discussion. Walker avoids the use of “a schema compiler” to generate “JAVA classes from an XML schema”. However, Walker still uses a regular/common Java compiler “*for compiling the code expressions to create the executable application*” as recited in claims (e.g., [0003], [0019]-[0021], using a common/regular Java compiler to compile “book store” Java classes to create a “book store” executable application).

Accordingly, the combination of the Walker and Bhansali is proper and these references fully teach the claimed limitations as set forth in details in the previous Office action mailed September 3, 2008.

b) “The rejection of claims 1-33 under 35 U.S.C. § 103(a) as being obvious over Walker in view of Bhansali should be withdrawn because the proposed combination fails to disclose each and every limitation of amended claim 9” (Remarks, pp. 16-18):

The limitations at issue: “*a parser for parsing the markup language file and providing the compiling system with detailed token information including non-code token information to the compiling system, wherein the markup language file is associated with at least one code-behind file*” (claim 9, lines 4-7, emphasis added).

The examiner respectfully disagrees with Applicants’ assertions. Walker explicitly teaches:

a parser for parsing the markup language file (e.g., FIG. 5, blocks 505-515 and related text, [0002], [0020]) and

providing the compiling system with detailed token information including non-code token information to the compiling system (e.g., page 2, an XML representation of a book store with tags, property/attributes, [0022]-[00023]);

wherein the markup language file is associated with at least one code-behind file (e.g., [0004], application programming interfaces (API) files associated with the XML file for converting between Java classes and said XML file; [0036]-[0038], XmlReaderWriter interface file).

In conclusion, the examiner respectfully maintains ground of the 35 USC §103 rejection over claims 1-9, 11-14, 16-26, and 28-33.

Claim Rejections – 35 USC §103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 1-9, 11-14, 16-26, and 28-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Walker (art of record, US Patent Publication No. 2004/0015840 A1) in view of Bhansali (art of record, US Patent Publication No. 2002/0169999 A1).

Claim 1:

Walker discloses *a compiling system embodied on a computer readable storage medium for compiling a markup language file into an executable application, the compiling system comprising:*

a parser for parsing the markup language file (e.g., FIG. 5, blocks 505-515 and related text, [0002], [0020]) and

providing the compiling system with detailed token information (e.g., page 2, an XML representation of a book store with tags, property/attributes, [0022]-[00023]);

a code generator for generating code expressions based on the token information (e.g., FIG. 5, blocks 535-565 and related text, [0003]-[0005], [0112]-[0114]),

wherein the code expressions represent the markup file as a class (e.g., page 2-3, Java class of book store, [0024]-[0028]); and

a compiler for compiling the code expressions to create the executable application (e.g., [0003], [0019]-[0021], using a common/regular Java compiler to compile "book store" classes to create a "book store" executable application).

Walker does not explicitly *generating a language-independent tree of code expressions.*

However, in an analogous art, Bhansali further discloses:

code expressions as a class (e.g., FIG. 9, code expressions as Input Stream 900, [0115]-[0117], [0121]-[0124]);

generating a language-independent tree of code expressions (e.g., [0018]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Bhansali's teaching into Walker's teaching. One would have been motivated to do so to translate a program into intermediate language code representation and into multiple a subsequent language code as suggested by Walker (e.g., [0017], [0020]-[0025]).

Claim 2:

The rejection of claim 1 is incorporated. Walker discloses *the detailed token information comprises a tag (e.g., [0038]-[0049]).*

Claim 3:

The rejection of claim 1 is incorporated. Walker discloses *the detailed token information comprises a property or event* (e.g., [0022]-[0026]).

Claim 4:

The rejection of claim 1 is incorporated. Walker discloses *the detailed token information comprises a user code snippet* (e.g., [0016]-[0021]).

Claim 5:

The rejection of claim 1 is incorporated. Walker discloses *the markup language file is associated with at least one code-behind file* (e.g., [0002]-[0005], application programming interfaces (API) files associated with the XML file for converting between Java classes and said XML file; [0036]-[0038], XmlReaderWriter interface file).

Claim 6:

The rejection of claim 5 is incorporated. Walker discloses *the compiler is configured to compile the markup language file and the code-behind file* (e.g., [0014]-[0021]).

Claim 7:

The rejection of claim 1 is incorporated. Walker discloses *the executable application is an intermediate language application* (e.g., [0039]-[0044]).

Claim 8:

The rejection of claim 1 is incorporated. Walker discloses *a binary file generator for generating a binary file from non-code token information, wherein the binary file contains one record for each non-code token* (e.g., [0060]-[0072]).

Claim 9:

Walker discloses *a compiling system embodied on a computer readable storage medium for compiling a markup language file into an executable application, the compiling system comprising:*

a parser for parsing the markup language file (e.g., FIG. 5, blocks 505-515 and related text, [0002], [0020]) and

providing the compiling system with detailed token information including non-code token information to the compiling system (e.g., page 2, an XML representation of a book store with tags, property/attributes, [0022]-[00023]),

wherein the markup language file is associated with at least one code-behind file (e.g., [0079]-[0088]; [0004], application programming interfaces (API) files associated with the XML file for converting between Java classes and said XML file; [0036]-[0038], XmlReaderWriter interface file);

a binary file generator for generating a binary file from non-code token information, wherein the binary file contains one record for each non-code token (e.g., [0088]);

a code generator for generating a language-independent code expression that represents the markup language file as a class (e.g., page 2-3, Java class of book store, [0024]-[0028]); and

an application generator for compiling the code files into an executable application (e.g., [0105]-[0118]; [0019]-[0021], using a Java compiler to compile "book store" classes to create a "book store" executable application).

Walker does not explicitly *generating a language-independent tree of code expressions*.

However, in an analogous art, Bhansali further discloses:

code expressions as a class (e.g., FIG. 9, code expressions as Input Stream 900, [0115]-[0117], [0121]-[0124]);

generating a language-independent tree of code expressions (e.g., [0018]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Bhansali's teaching into Walker's teaching. One would have been motivated to do so to translate a program into intermediate language code representation and into multiple a subsequent language code as suggested by Walker (e.g., [0017], [0020]-[0025]).

Claim 11:

The rejection of claim 9 is incorporated. Walker discloses *the application generator combines the binary files into a single resource* (e.g., [0098]-[0114])

Claim 12:

The rejection of claim 9 is incorporated. Walker discloses *the detailed token information comprises a tag* (e.g., [0078]-[0092]).

Claim 13:

The rejection of claim 9 is incorporated. Walker discloses *the detailed token information comprises a property or event* (e.g., [0044]-[0056]).

Claim 14:

The rejection of claim 9 is incorporated. Walker discloses *the detailed token information comprises a user code snippet* (e.g., [0061]-[0072]).

Claim 16:

The rejection of claim 9 is incorporated. Walker discloses *the compiling system is configured to compile the markup language file and the code-behind file* (e.g., [0016]-[0024]).

Claim 17:

Walker discloses *a method for compiling a markup language file into an executable application, the method comprising:*

receiving a markup language file; parsing the markup language file and providing a compiling system with detailed token information (e.g., FIG. 5, blocks 505-515 and related text, [0002], [0020]; page 2, an XML representation of a book store with tags, property/attributes, [0022]-[00023]);

generating code expressions based on the token information (e.g., FIG. 5, blocks 535-565 and related text, [0003]-[0005], [0112]-[0114]),
wherein the code expressions represent the markup language file as a class (e.g., page 2-3, Java class of book store, [0024]-[0028]); and
compiling the code expressions to create the executable application (e.g., [0003], [0019]-[0021], using a Java compiler to compile “book store” classes to create a “book store” executable application).

Walker does not explicitly *generating a language-independent tree of code expressions*.

However, in an analogous art, Bhansali further discloses:

code expressions as a class (e.g., FIG. 9, code expressions as Input Stream 900, [0115]-[0117], [0121]-[0124]);

generating a language-independent tree of code expressions (e.g., [0018]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Bhansali's teaching into Walker's teaching. One would have been motivated to do so to translate a program into intermediate language code representation and into multiple a subsequent language code as suggested by Walker (e.g., [0017], [0020]-[0025]).

Claim 18:

The rejection of claim 17 is incorporated. Walker discloses *receiving a code-behind file* (e.g., [0033]-[0042]).

Claim 19:

The rejection of claim 18 is incorporated. Walker discloses *compiling the markup language file and the code-behind file* (e.g., [0049]-[0058]).

Claim 20:

The rejection of claim 17 is incorporated. Walker discloses *providing a tag as detailed token information* (e.g., [0003]-[0011]).

Claim 21:

The rejection of claim 17 is incorporated. Walker discloses *providing a property or event as the detailed token information* (e.g., [0026]-[0034]).

Claim 22:

The rejection of claim 17 is incorporated. Walker discloses *providing a user code snippet as the detailed token information* (e.g., [0051]-[0064]).

Claim 23:

The rejection of claim 17 is incorporated. Walker discloses *receiving a command to create an intermediate language application* (e.g., [0028]-[0040]).

Claim 24:

The rejection of claim 17 is incorporated. Walker discloses *generating a binary file from non-code token information, wherein the binary file contains one record for each non-code token* (e.g., [0016]-[0022], [0026]-[0049]).

Claim 25:

Claim 25 is a computer readable storage medium version, which recite(s) the same limitations as those of claim 17, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claim(s), it also teaches all of the limitations of claim 25.

Claim 26:

Walker discloses *a method for compiling a markup language file into an executable application, the method comprising:*

parsing the markup language file and providing the compiling system with detailed token information including non-code token information (e.g., FIG. 5, blocks 505-515 and related text, [0002], [0020]);

generating a binary file from the non-code token information, wherein the binary file contains one record for each non-code token (e.g., [0088]);

generating a language-independent code expression that represents the markup language file as a class (e.g., page 2-3, Java class of book store, [0024]-[0028]; [0003], [0019]); and

compiling the code expressions into an executable application (e.g., [0057]-[0069]; [0019]-[0021], using a Java compiler to compile “book store” classes to create a “book store” executable application).

Walker does not explicitly *generating a language-independent tree of code expressions*.

However, in an analogous art, Bhansali further discloses:

code expressions as a class (e.g., FIG. 9, code expressions as Input Stream 900, [0115]-[0117], [0121]-[0124]);

generating a language-independent tree of code expressions (e.g., [0018]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Bhansali's teaching into Walker's teaching. One would have been motivated to do so to translate a program into intermediate language code representation and into multiple a subsequent language code as suggested by Walker (e.g., [0017], [0020]-[0025]).

Claim 28:

The rejection of claim 26 is incorporated. Walker discloses *combining the binary files into a single resource (e.g., [0061]-[0073]).*

Claim 29:

The rejection of claim 26 is incorporated. Walker discloses *providing a tag as the detailed token information (e.g., [0094]-[0111]).*

Claim 30:

The rejection of claim 26 is incorporated. Walker discloses *providing a property or event as the detailed token information* (e.g., [0114]-[0118]).

Claim 31:

The rejection of claim 26 is incorporated. Walker discloses *providing a user code snippet as the detailed token information* (e.g., [0019]-[0026]).

Claim 32:

The rejection of claim 26 is incorporated. Walker discloses *receiving at least one code-behind file associated with the markup language file and compiling both the code-behind file and the markup language file* (e.g., [0025]-[0041]).

Claim 33:

Claim 33 is a computer readable storage medium version, which recite(s) the same limitations as those of claim 26, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claim(s), it also teaches all of the limitations of claim 33.

Conclusion

9. THIS ACTION IS MADE FINAL. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action.

Art Unit: 2192

In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

10. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone/fax numbers are (571) 272 8570 and (571) 273 8570, respectively. The examiner can normally be reached on every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Thuy Dao/

Examiner, Art Unit 2192

/Tuan Q. Dam/

Supervisory Patent Examiner, Art Unit 2192